WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Tuesday, March 02, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT; PLUR=YES; OP=ADJ	
☐	L14	L13 and l11	2
☐	L13	L10 and monitor$	32
☐	L12	L11 and interac$	3
☐	L11	L10 and l4	4
☐	L10	L9 and l1 and l2	45
☐	L9	L8 and l5	10724
☐	L8	L6 and (compos$ or control$)	14316
☐	L7	L6 and (compos$ or control$)	35
☐	L6	gui or graphical user interface	15044
☐	L5	www or internet or network	289788
☐	L4	nntp and l2	15
☐	L3	L2 and loom	0
☐	L2	irc or internet relay chat	2169
☐	L1	client and server	20354

END OF SEARCH HISTORY

h e b b cg b chh e b g f c e h

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs
Generate OACS

Search Results - Record(s) 1 through 4 of 4 returned.

☑ 1. Document ID: US 6633850 B1

L11: Entry 1 of 4 File: USPT Oct 14, 2003

US-PAT-NO: 6633850
DOCUMENT-IDENTIFIER: US 6633850 B1

TITLE: Background advertising system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☑ 2. Document ID: US 6633835 B1

L11: Entry 2 of 4 File: USPT Oct 14, 2003

US-PAT-NO: 6633835
DOCUMENT-IDENTIFIER: US 6633835 B1

TITLE: Prioritized data capture, classification and filtering in a network monitoring environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☑ 3. Document ID: US 6401118 B1

L11: Entry 3 of 4 File: USPT Jun 4, 2002

US-PAT-NO: 6401118
DOCUMENT-IDENTIFIER: US 6401118 B1

TITLE: Method and computer program product for an online monitoring search engine

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 4. Document ID: US 6205432 B1

L11: Entry 4 of 4 File: USPT Mar 20, 2001

US-PAT-NO: 6205432
DOCUMENT-IDENTIFIER: US 6205432 B1

TITLE: Background advertising system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Term	Documents
(4 AND 10).USPT.	4
(L10 AND L4).USPT.	4

Display Format: TI Change Format

Previous Page Next Page Go to Doc#

h e b b g e e e f e b g ef b e

First Hit Fwd Refs

End of Result Set

Generate Collection Print

L3: Entry 1 of 1 File: USPT May 19, 1998

DOCUMENT-IDENTIFIER: US 5754939 A
TITLE: System for generation of user profiles for a system for customized electronic identification of desirable objects

Brief Summary Text (19):
The preferred embodiment of the system for customized electronic identification of desirable objects operates in an electronic media environment for accessing these target objects, which may be news, electronic mail, other published documents, or product descriptions. The system in its broadest construction comprises three conceptual modules, which may be separate entities distributed across many implementing systems, or combined into a lesser subset of physical entities. The specific embodiment of this system disclosed herein illustrates the use of a first module which automatically constructs a "target profile" for each target object in the electronic media based on various descriptive attributes of the target object. A second module uses interest feedback from users to construct a "target profile interest summary" for each user, for example in the form of a "search profile set" consisting of a plurality of search profiles, each of which corresponds to a single topic of high interest for the user. The system further includes a profile processing module which estimates each user's interest in various target objects by reference to the users' target profile interest summaries, for example by comparing the target profiles of these target objects against the search profiles in users' search profile sets, and generates for each user a customized rank-ordered listing of target objects most likely to be of interest to that user. Each user's target profile interest summary is automatically updated on a continuing basis to reflect the user's changing interests.

Brief Summary Text (22):
There are a number of variations on the theme of developing and using profiles for article retrieval, with the basic implementation of an on-line news clipping service representing the preferred embodiment of the invention. Variations of this basic system are disclosed and comprise a system to filter electronic mail, an extension for retrieval of target objects such as purchasable items which may have more complex descriptions, a system to automatically build and alter menuing systems for browsing and searching through large numbers of target objects, and a system to construct virtual communities of people with common interests. These intelligent filters and browsers are necessary to provide a truly passive, intelligent system interface. A user interface that permits intuitive browsing and filtering represents for the first time an intelligent system for determining the affinities between users and target objects. The detailed, comprehensive target profiles and user-specific target profile interest summaries enable the system to provide responsive routing of specific queries for user information access. The information maps so produced and the application of users' target profile interest summaries to predict the information consumption patterns of a user allows for pre-caching of data at locations on the data communication network and at times that minimize the traffic flow in the communication network to thereby efficiently provide the desired information to the user and/or conserve valuable storage space by only storing those target objects (or segments thereof) which are relevant to the user's interests.

Detailed Description Text (31):
(f.) number of stars granted by a second critic,

Detailed Description Text (49):
(a.) first two digits of zip code (textual),

Detailed Description Text (50):
(b.) first three digits of zip code (textual),

Detailed Description Text (73):
First, define the distance between two values of a given attribute according to whether the attribute is a numeric, associative, or textual attribute. If the attribute is numeric, then the distance between two values of the attribute is the absolute value of the difference between the two values. (Other definitions are also possible: for example, the distance between prices p1 and p2 might be defined by .vertline.(p1-p2).vertline./(max(p1,p2)+1), to recognize that when it comes to customer interest, $5000 and $5020 are very similar, whereas $3 and $23 are not.) If the attribute is associative, then its value V may be decomposed as described above into a collection of real numbers, representing the association scores between the target object in question and various ancillary objects. V may therefore be regarded as a vector with components V.sub.1, V.sub.2, V.sub.3, etc., representing the association scores between the object and ancillary objects 1, 2, 3, etc., respectively. The distance between two vector values V and U of an associative attribute is then computed using the angle distance measure, arccos (VU.sup.t /sqrt((Vv.sup.t)(UU.sup.t)). (Note that the three inner products in this expression have the form XY.sup.t =X.sub.1 Y.sub.1 +X.sub.2 Y.sub.2 +X.sub.3 Y.sub.3 + . . . , and that for efficient computation, terms of the form X.sub.i Y.sub.i may be omitted from this sum if either of the scores X.sub.i and Y.sub.i is zero.) Finally, if the attribute is textual, then its value V may be decomposed as described above into a collection of real numbers, representing the scores of various word n-grams or character n-grams in the text. Then the value V may again be regarded as a vector, and the distance between two values is again defined via the angle distance measure. Other similarity metrics between two vectors, such as the dice measure, may be used instead. It happens that the obvious alternative metric, Euclidean distance, does not work well: even similar texts tend not to overlap substantially in the content words they use, so that texts encountered in practice are all substantially orthogonal to each other, assuming that TF/IDF scores are used to reduce the influence of non-content words. The scores of two words in a textual attribute vector may be correlated; for example, "Kennedy" and "JFK" tend to appear in the same documents. Thus it may be advisable to alter the text somewhat before computing the scores of terms in the text, by using a synonym dictionary that groups together similar words. The effect of this optional pre-alteration is that two texts using related words are measured to be as similar as if they had actually used the same words. One technique is to augment the set of words actually found in the article with a set of synonyms or other words which tend to co-occur with the words in the article, so that "Kennedy" could be added to every article that mentions "JFK." Alternatively, words found in the article may be wholly replaced by synonyms, so that "JFK" might be replaced by "Kennedy" or by "John F. Kennedy" wherever it appears. In either case, the result is that documents about Kennedy and documents about JFK are adjudged similar. The synonym dictionary may be sensitive to the topic of the document as a whole; for example, it may recognize that "crane" is likely to have a different synonym in a document that mentions birds than in a document that mentions construction. A related technique is to replace each word by its morphological stem, so that "staple", "stapler", and "staples" are all replaced by "staple." Common function words ("a", "and", "the" . . .) c an influence the calculated similarity of texts without regard to their topics, and so are typically removed from the text before the scores of terms in the text are computed. A more general approach to recognizing synonyms is to use a revised measure of the distance between textual attribute vectors V and U, namely

arccos(AV(AU).sup.t /sqrt (AV(AV).sup.t AU(AU).sup.t), where the matrix A is the dimensionality-reducing linear transformation (or an approximation thereto) determined by collecting the vector values of the textual attribute, for all target objects known to the system, and applying singular value decomposition to the resulting collection. The same approach can be applied to the vector values of associative attributes. The above definitions allow us to determine how close together two target objects are with respect to a single attribute, whether numeric, associative, or textual. The distance between two target objects X and Y with respect to their entire multi-attribute profiles P.sub.x and P.sub.y is then denoted d(X,Y) or d(P.sub.x, P.sub.y) and defined as:

Detailed Description Text (80):
+2 if the second page is viewed,

Detailed Description Text (82):
+2 if more than 30 seconds was spent viewing the document,

Detailed Description Text (85):
If the target objects are electronic mail messages, interest points might also be added in the case of a particularly lengthy or particularly prompt reply. If the target objects are purchasable goods, interest points might be added for target objects that the user actually purchases, with further points in the case of a large-quantity or high-price purchase. In any domain, further points might be added for target objects that the user accesses early in a session, on the grounds that users access the objects that most interest them first. Other potential sources of passive feedback include an electronic measurement of the extent to which the user's pupils dilate while the user views the target object or a description of the target object. It is possible to combine active and passive feedback. One option is to take a weighted average of the two ratings. Another option is to use passive feedback by default, but to allow the user to examine and actively modify the passive feedback score. In the scenario above, for instance, an uninteresting article may sometimes remain on the display device for a long period while the user is engaged in unrelated business; the passive feedback score is then inappropriately high, and the user may wish to correct it before continuing. In the preferred embodiment of the invention, a visual indicator, such as a sliding bar or indicator needle on the user's screen, can be used to continuously display the passive feedback score estimated by the system for the target object being viewed, unless the user has manually adjusted the indicator by a mouse operation or other means in order to reflect a different score for this target object, after which the indicator displays the active feedback score selected by the user, and this active feedback score is used by the system instead of the passive feedback score. In a variation, the user cannot see or adjust the indicator until just after the user has finished viewing the target object. Regardless how a user's feedback is computed, it is stored long-term as part of that user's target profile interest summary.

Detailed Description Text (106):
4) Sequential hybrid method. First apply the k-means procedure to do 1a, so that articles are labeled by cluster based on which user read them, then use supervised clustering (maximum likelihood discriminant methods) using the word frequencies to do the process of method 2a described above. This tries to use knowledge of who read what to do a better job of clustering based on word frequencies. One could similarly combine the methods 1b and 2b described above.

Detailed Description Text (107):
Hierarchical clustering of target objects is often useful. Hierarchical clustering produces a tree which divides the target objects first into two large clusters of roughly similar objects; each of these clusters is in turn divided into two or more smaller clusters, which in turn are each divided into yet smaller clusters until the collection of target objects has been entirely divided into "clusters"

consisting of a single object each, as diagrammed in FIG. 8 In this diagram, the node d denotes a particular target object d, or equivalently, a single-member cluster consisting of this target object. Target object d is a member of the cluster (a, b, d), which is a subset of the cluster (a, b, c, d, e, f), which in turn is a subset of all target objects. The tree shown in FIG. 8 would be produced from a set of target objects such as those shown geometrically in FIG. 7. In FIG. 7, each letter represents a target object, and axes x1 and x2 represent two of the many numeric attributes on which the target objects differ. Such a cluster tree may be created by hand, using human judgment to form clusters and subclusters of simlar objects, or may be created automatically in either of two standard ways: top-down or bottom-up. In top-down hierarchical clustering, the set of all target objects in FIG. 7 would be divided into the clusters (a, b, c, d, e, f) and (g, h, i j, k). The clustering algorithm would then be reapplied to the target objects in each cluster, so that the cluster (g, h, i, j, k) is subpartitioned into the clusters (g, k) and (h, i, j), and so on to arrive at the tree shown in FIG. 8. In bottom-up hierarchical clustering, the set of all target objects in FIG. 7 would be grouped into numerous small clusters, namely (a, b), d, (c, f), e, (g, k), (h, i), andj. These clusters would then themselves be grouped into the larger clusters (a, b, d), (c, e, f), (g, k), and (h, i, j), according to their cluster profiles. These larger clusters would themselves be grouped into (a, b, c, d, e, f) and (g, k, h, i, j), and so on until all target objects had been grouped together, resulting in the tree of FIG. 8. Note that for bottom-up clustering to work, it must be possible to apply the clustering algorithm to a set of existing clusters. This requires a notion of the distance between two clusters. The method disclosed above for measuring the distance between target objects can be applied directly, provided that clusters are profiled in the same way as target objects. It is only necessary to adopt the convention that a cluster's profile is the average of the target profiles of all the target objects in the cluster; that is, to determine the cluster's value for a given attribute, take the mean value of that attribute across all the target objects in the cluster. For the mean value to be well-defined, all attributes must be numeric, so it is necessary as usual to replace each textual or associative attribute with its decomposition into numeric attributes (scores), as described earlier. For example, the target profile of a single Woody Allen film would assign "Woody-Allen" a score of 1 in the "name-of-director" field, while giving "Federico-Fellini" and "Terence-Davies" scores of 0. A cluster that consisted of 20 films directed by Allen and 5 directed by Fellini would be profiled with scores of 0.8, 0.2, and 0 respectively, because, for example, 0.8 is the average of 20 ones and 5 zeros.

Detailed Description Text (121):
In some domains, complete profiles of target objects are not always easy to construct automatically. When target objects are wallpaper patterns, for example, an attribute such as "genre" (a single textual term such as "Art-Deco," "Children's," "Rustic," etc.) may be a matter ofjudgment and opinion, difficult to determine except by consulting a human. More significantly, if each wallpaper pattern has an associative attribute that records the positive or negative relevance feedback to that pattern from various human users (consumers), then all the association scores of any newly introduced pattern are initially zero, so that it is initially unclear what other patterns are similar to the new pattern with respect to the users who like them. Indeed, if this associative attribute is highly weighted, the initial lack of relevance feedback information may be difficult to remedy, due to a vicious circle in which users of moderate-to-high interest are needed to provide relevance feedback but relevance feedback is needed to identify users of moderate-to-high interest. Fortunately, however, it is often possible in principle to determine certain attributes of a new target object by extraordinary methods, including but not limited to methods that consult a human. For example, the system can in principle determine the genre of a wallpaper pattern by consulting one or more randomly chosen individuals from a set of known human experts, while to determine the numeric association score between a new wallpaper pattern and a particular user, it can in principle show the pattern to the that

user and obtain relevance feedback. Since such requests inconvenience people, however, it is important not to determine all difficult attributes this way, but only the ones that are most important for purposes of classifying the document. "Rapid profiling" is a method for selecting those numeric attributes that are most important to determine. (Recall that all attributes can be decomposed into numeric attributes, such as association scores or term scores.) First, a set of existing target objects that already have complete or largely complete profiles are clustered using a k-means algorithm. Next, each of the resulting clusters is assigned a unique identifying number, and each clustered target object is labeled with the identifying number of its cluster. Standard methods then allow construction of a single decision tree that can determine any target object's cluster number, with substantial accuracy, by considering the attributes of the target object, one at a time. Only attributes that can if necessary be determined for any new target object are used in the construction of this decision tree. To profile a new target object, the decision tree is traversed downward from its root as far as is desired. The root of the decision tree considers some attribute of the target object. If the value of this attribute is not yet known, it is determined by a method appropriate to that attribute; for example, if the attribute is the association score of the target object with user #4589, then relevance feedback (to be used as the value of this attribute) is solicited from user #4589, perhaps by the ruse of adding the possibly uninteresting target object to a set of objects that the system recommends to the user's attention, in order to find out what the user thinks of it. Once the root attribute is determined, the rapid profiling method descends the decision tree by one level, choosing one of the decision subtrees of the root in accordance with the determined value of the root attribute. The root of this chosen subtree considers another attribute of the target object, whose value is likewise determined by an appropriate method. The process c an be repeated to determine as many attributes as desired, by whatever methods are available, although it is ordinarily stopped after a small number of attributes, to avoid the burden of determining too many attributes.

Detailed Description Text (122):
It should be noted that the rapid profiling method can be used to identify important attributes in any sort of profile, and not just profiles of target objects. In particular, recall that the disclosed method for determining topical interest through similarity requires users as well as target objects to have profiles. New users, like new target objects, may be profiled or partially profiled through the rapid profiling process. For example, when user profiles include an associative attribute that records the user's relevance feedback on a 11 target objects in the system, the rapid profiling procedure can rapidly form a rough characterization of a new user's interests by soliciting the user's feedback on a small number of significant target objects, and perhaps also by determining a small n umber of other key attributes of the new user, by on-line queries, telephone surveys, or other means. Once the new user has been partially profiled in this way, the methods disclosed above predict that the new user's interests resemble the known interests of other users with similar profiles. In a variation, each user's user profile is subdivided into a set of long-term attributes, such as demographic characteristics, and a set of short-term attributes that help to identify the user's temporary desires and emotional state, such as the user's textual or multiple-choice answers to questions whose answers reflect the user's mood. A subset of the user's long-term attributes are determined when the user first registers with the system, through the use of a rapid profiling tree of long-term attributes. In addition, each time the user logs on to the system, a subset of the user's short-term attributes are additionally determined, through the use of a separate rapid profiling tree that asks about short-term attributes.

Detailed Description Text (136):
However, complete privacy and inaccessibility of user transactions and profile summary information would hinder implementation of the system for customized electronic identification of desirable objects and would deprive the user of many

of the advantages derived through the system's use of user-specific information. In many cases, complete and total privacy is not desired by all parties to a transaction. For example, a buyer may desire to be targeted for certain mailings that describe products that are related to his or her interests, and a seller may desire to target users who are predicted to be interested in the goods and services that the seller provides. Indeed, the usefulness of the technology described herein is contingent upon the ability of the system to collect and compare data about many users and many target objects. A compromise between total user anonymity and total public disclosure of the user's search profiles or target profile interest summary is a pseudonym. A pseudonym is an artifact that allows a service provider to communicate with users and build and accumulate records of their preferences over time, while at the same time remaining ignorant of the users' true identities, so that users can keep their purchases or preferences private. A second and equally important requirement of a pseudonym system is that it provide for digital credentials, which are used to guarantee that the user represented by a particular pseudonym has certain properties. These credentials may be granted on the basis of result of activities and transactions conducted by means of the system for customized electronic identification of desirable objects, or on the basis of other activities and transactions conducted on the network N of the present system, on the basis of users' activities outside of network N. For example, a service provider may require proof that the purchaser has sufficient funds on deposit at his/her bank, which might possibly not be on a network, before agreeing to transact business with that user. The user, therefore, must provide the service provider with proof of funds (a credential) from the bank, while still not disclosing the user's true identity to the service provider.

Detailed Description Text (138):
1. The first function of the proxy server is to bidirectionally transfer communications between user U and other entities such as information servers (possibly including the proxy server itself) and/or other users. Specifically, letting S denote the server that is directly associated with user U's client processor, the proxy server communicates with server S (and thence with user U), either through anonymizing mix paths that obscure the identity of server S and user U, in which case the proxy server knows user U only through a secure pseudonym, or else through a conventional virtual point-to-point connection, in which case the proxy server knows user U by user U's address at server S, which address may be regarded as a non-secure pseudonym for user U.

Detailed Description Text (139):
2. A second function of the proxy server is to record user-specific information associated with user U. This user-specific information includes a user profile and target profile interest summary for user U, as well as a list of access control instructions specified by user U, as described below, and a set of one-time return addresses provided by user U that can be used to send messages to user U without knowing user U's true identity. All of this user-specific information is stored in a database that is keyed by user U's pseudonym (whether secure or non-secure) on the proxy server.

Detailed Description Text (151):
In our implementation, a pseudonym is a data record consisting of two fields. The first field specifies the address of the proxy server at which the pseudonym is registered. The second field contains a unique string of bits (e.g., a random binary number) that is associated with a particular user; credentials take the form of public-key digital signatures computed on this number, and the number itself is issued by a pseudonym administering server Z, as depicted in FIG. 2, and detailed I n a generic form in the paper by D. Chaum and J. H. Evertse, titled "A secure and privacy-protecting protocol for transmitting personal information between organizations." It is possible to send information to the user holding a given pseudonym, by enveloping the information in a control message that specifies the pseudonym and is addressed to the proxy server that is named in the first field of

the pseudonym; the proxy server may forward the information to the user upon receipt of the control message.

Detailed Description Text (173):
In general, the user requests access to a particular target object or menu of target objects; once the corresponding file has been transmitted to the user's client processor, the user views its contents and makes another such request, and so on. Each request may take many seconds to satisfy, due to retrieval and transmission delays. However, to the extent that the sequence of requests is predictable, the system for customized electronic identification of desirable objects can respond more quickly to each request, by retrieving or starting to retrieve the appropriate files even before the user requests them. This early retrieval is termed "pre-fetching of files."

Detailed Description Text (176):
Pre-fetching exhibits a cost-benefit tradeoff. Let t denote the approximate number of minutes that pre-fetched files are retained in local storage (before they are deleted to make room for other pre-fetched files). If the system elects to pre-fetch a file corresponding to a target object X, then the user benefits from a fast response at no extra cost, provided that the user explicitly requests target object X soon thereafter. However, if the user does not request target object X within t minutes of the pre-fetch, then the pre-fetch was worthless, and its cost is an added cost that must be borne (directly or indirectly) by the user. The first scenario therefore provides benefit at no cost, while the second scenario incurs a cost at no benefit. The system tries to favor the first scenario by pre-fetching only those files that the user will access anyway. Depending on the user's wishes, the system may pre-fetch either conservatively, where it controls costs by pre-fetching only files that the user is extremely likely to request explicitly (and that are relatively cheap to retrieve), or more aggressively, where it also pre-fetches files that the user is only moderately likely to request explicitly, thereby increasing both the total cost and (to a lesser degree) the total benefit to the user.

Detailed Description Text (185):
The difficult task is for proxy server S, each time it retrieves a file F in response to a request, to identify the files G1 . . . Gk that should be triggered by the request for file F and pre-fetched immediately. Proxy server S employs a cost-benefit analysis, performing each pre-fetch whose benefit exceeds a user-determined multiple of its cost; the user may set the multiplier low for aggressive prefetching or high for conservative prefetching. These pre-fetches may be performed in parallel. The benefit of pre-fetching file Gi immediately is defined to be the expected number of seconds saved by such a pre-fetch, as compared to a situation where Gi is left to be retrieved later (either by a later pre-fetch, or by the user's request) if at all. The cost of pre-fetching file Gi immediately is defined to be the expected cost for proxy server S to retrieve file Gi, as determined for example by the network locations of server S and file Gi and by information provider charges, times 1 minus the probability that proxy server S will have to retrieve file Gi within t minutes (to satisfy either a later pre-fetch or the user's explicit request) if it is not pre-fetched now.

Detailed Description Text (242):
Algorithms for constructing multicast trees have either been ad-hoc, as is the case of the Deering, et al. Internet multicast tree, which adds clients as they request service by grafting them into the existing tree, or by construction of a minimum cost spanning tree. A distributed algorithm for creating a spanning tree (defined as a tree that connects, or "spans," all nodes of the graph) on a set of Ethernet bridges was developed by Radia Perlman ("Interconnections: Bridges and Routers," Radia Perlman, Addison-Wesley, 1992). Creating a minimal-cost spanning tree for a graph depends on having a cost model for the arcs of the graph (corresponding to communications 1 inks in the communications network). In the case of Ethernet

bridges, the default cost (more complicated costing models for path costs are discussed on pp. 72-73 of Perlman) is calculated as a simple distance measure to the root; thus the spanning tree minimizes the cost to the root by first electing a unique root and then constructing a spanning tree based on the distances from the root. In this algorithm, the root is elected by recourse to a numeric ID contained in "configuration messages": the server w hose ID has minimum numeric value is chosen as the root. Several problems exist with this algorithm in general. First, the method of using an ID does not necessarily select the best root for the nodes interconnected in the tree. Second, the cost model is simplistic.

Detailed Description Text (243):
We first show how to use the similarity-based methods described above to select the servers most interested in a group of target objects, herein termed "core servers" for that group. Next we show how to construct an unrooted multicast tree that can be used to broadcast files to these core servers. Finally, we show how files corresponding to target objects are actually broadcast through the multicast tree at the initiative of a client, and how these files are later retrieved from the core servers when clients request them.

Detailed Description Text (269):
In addition to global request messages, another type of message that may be transmitted to any proxy server S is termed a "query message." When transmitted to a proxy server, a query message causes a reply to be sent to the originator of the message; this reply will contain an answer to a given query Q if any of the servers in a given multicast tree MT(C) are able to answer it, and will otherwise indicate that no answer is available. The query and the cluster C are named in the query message. In addition, the query message contains a field S.sub.last which is unspecified except under certain circumstances described below, when it names a specific core server. When a proxy server S receives a message M that is marked as a query message, it acts as follows:1. Proxy server S sets A.sub.r to be the return address for the client or server that transmitted message M to server S. A.sub.r may be either a network address or a pseudonymous address 2. If proxy server S is not a core server for cluster C, it retrieves its locally stored list of nearby core servers for topic C, selects from this list a nearby core server S', and transmits a copy of the locate message M over a virtual point-to-point connection to core server S'. If this transmission fails, proxy server S repeats the procedure with other core servers on its list. Upon receiving a reply, it forwards this reply to address A.sub.r 3. If proxy server S is a core server for cluster C, and it is able to answer query Q using locally stored information, then it transmits a "positive" reply to A.sub.r containing the answer. 4. If proxy server S is a core server for topic C, but it is unable to answer query Q using locally stored information, then it carries out a parallel depth-first search by executing the following steps: (a) Set L to be the empty list. (b) Retrieve the locally stored subtree of MT(C). For each server Si directly linked to S.sub.curr in this subtree, other than S.sub.last (if specified), add the ordered pair (Si S) to the list L. (c) If L is empty, transmit a "negative" reply to address A.sub.r saying that server S cannot locate an answer to query Q, and terminate the execution of step 4; otherwise proceed to step (d). (d) Select a list L1 of one or more server pairs (Ai, Bi) from the list L. For each server pair (Ai, Bi) on the list L1, form a locate message M(Ai, Bi), which is a copy of message M whose S.sub.last field has been modified to specify Bi and transmit this message M(Ai Bi) to server Ai over a virtual point-to-point connection. (e) For each reply received (by S) to a message sent in step (d), act as follows: (I) If a "positive" reply arrives to a locate message M(Ai, Bi), then forward this reply to A.sub.r and terminate step 4, immediately. (ii) If a "negative" reply arrives to a locate message M(Ai, Bi), then remove the pair (Ai, Bi) from the list L1. (iii) If the message M(Ai, Bi) could not be successfully delivered to Ai, then remove the pair (Ai, Bi) from the list L1, and add the pair (Ci, Ai) to the list L1 for each Ci other than Bi that is directly linked to Ai in the locally stored subtree of MT(C). (f) Once L1 no longer contains any pair (Ai, Bi) for which a message M(Ai, Bi) has been sent, or after a fixed

period of time has elapsed, return to step (c).

Detailed Description Text (285):
Notice that the above variation attempts to match clusters of search profiles with similar clusters of articles. Since this is a symmetrical problem, it may instead be given a symmetrical solution, as the following more general variation shows. At some point before the matching process commences, all the news articles to be considered are clustered into a hierarchical tree, termed the "target profile cluster tree," and the search profiles of all users to be considered are clustered into a second hierarchical tree, termed the "search profile cluster tree." The following steps serve to find all matches between individual target profiles from any target profile cluster tree and individual search profiles from any search profile cluster tree: 1. For each child subtree S of the root of the search profile cluster tree (or, let S be the entire search profile cluster tree if it contains only one search profile): 2. Compute the cluster profile P.sub.s to be the average of all search profiles in subtree S 3. For each subdluster (child subtree) T of the root of the target profile cluster tree (or, let T be the entire target profile cluster tree if it contains only one target profile): 4. Compute the cluster profile P.sub.T to be the average of all target profiles in subtree T 5. Calculate d(P.sub.s, P.sub.T, the distance between P.sub.s and P.sub.T 6. If d(P.sub.s P.sub.T) < t, a threshold, 7. If S contains only one search profile and T contains only one target profile, decl are a match between that search profile and that target profile, 8. otherwise recurse to step 1 to find all matches between search profiles in tree S and target profiles in tree T.

Detailed Description Text (288):
Once the profile correlation step is completed for a selected user or group of users, at step 1104 the profile processing module 203 stores a list of the identified articles for presentation to each user. At a user's request, the profile processing system 203 retrieves the generated list of relevant articles and presents this list of titles of the selected articles to the user, who can then select at step 1105 any article for viewing. (If no titles are available, then the first sentence(s) of each article can be used.) The list of article titles is sorted according to the degree of similarity of the article's target profile to the most similar search profile in the user's search profile set. The resulting sorted list is either transmitted in real time to the user client processor C.sub.1, if the user is present at their client processor C.sub.1, or can be transmitted to a user's mailbox, resident on the user's client processor C.sub.1 or stored within the server S.sub.2 for later retrieval by the user; other methods of transmission include facsimile transmission of the printed list or telephone transmission by means of a text-to-speech system. The user can then transmit a request by computer, facsimile, or telephone to indicate which of the identified articles the user wishes to review, if any. The user can still access all articles in any information server S.sub.4 to which the user has authorized access, however, those lower on the generated list are simply further from the user's interests, as determined by the user's search profile set. The server S.sub.2 retrieves the article from the local data storage medium or from an information server S.sub.4 and presents the article one screen at a time to the user's client processor C.sub.1. The user can at any time select another article for reading or exit the process.

Detailed Description Text (290):
The user's search profile set generator 202 at step 1107 monitors which articles the user reads, keeping track of how many pages of text are viewed by the user, how much time is spent viewing the article, and whether all pages of the article were viewed. This information can be combined to measure the depth of the user's interest in the article, yielding a passive relevance feedback score, as described earlier. Although the exact details depend on the length and nature of the articles being searched, a typical formula might be: measure of article attractiveness =0.2 if the second page is accessed +0.2 if all pages are accessed +0.2 if more than 30 seconds was spent on the article +0.2 if more than one minute was spent on the

article +0.2 if the minutes spent in the article are greater than half the number of pages.

Detailed Description Text (298):
The filtering technology of the news clipping service is not limited to news articles provided by a single source, but may be extended to articles or target objects collected from any number of sources. For example, rather than identifying new news articles of interest, the technology may identify new or updated World Wide Web pages of interest. In a second application, termed "broadcast clipping," where individual users desire to broadcast messages to all interested users, the pool of news articles is replaced by a pool of messages to be broadcast, and these messages are sent to the broadcast-clipping-service subscribers most interested in them. In a third application, the system scans the transcripts of all real-time spoken or written discussions on the network that are currently in progress and designated as public, and employs the news-clipping technology to rapidly identify discussions that the user may be interested in joining, or to rapidly identify and notify users who may be interested in joining an ongoing discussion. In a fourth application, the method is used as a post-process that filters and ranks in order of interest the many target objects found by a conventional database search, such as a search for all homes selling for under $200,000 in a given area, for all 1994 news articles about Marcia Clark, or for all Italian-language films. In a fifth application, the method is used to filter and rank the links in a hypertext document by estimating the user's interest in the document or other object associated with each link. In a sixth application, paying advertisers, who may be companies or individuals, are the source of advertisements or other messages, which take the place of the news articles in the news clipping service. A consumer who buys a product is deemed to have provided positive relevance feedback on advertisements for that product, and a consumer who buys a product apparently because of a particular advertisement (for example, by using a coupon clipped from that advertisement) is deemed to have provided particularly high relevance feedback on that advertisement. Such feedback may be communicated to a proxy server by the consumer's client processor (if the consumer is making the purchase electronically), by the retail vendor, or by the credit-card reader (at the vendor's establishment) that the consumer uses to pay for the purchase. Given a database of such relevance feedback, the disclosed technology is then used to match advertisements with those users who are most interested in them; advertisements selected for a user are presented to that user by any one of several means, including electronic mail, automatic display on the users screen, or printing them on a printer at a retail establishment where the consumer is paying for a purchase. The threshold distance used to identify interest may be increased for a particular advertisement, causing the system to present that advertisement to more users, in accordance with the amount that the advertiser is willing to pay.

Detailed Description Text (320):
A hierarchical cluster tree imposes a useful organization on a collection of target objects. The tree is of direct use to a user who wishes to browse through all the target objects in the tree. Such a user may be exploring the collection with or without a well-specified goal. The tree's division of target objects into coherent clusters provides an efficient method whereby the user can locate a target object of interest. The user first chooses one of the highest level (largest) clusters from a menu, and is presented with a menu listing the subdlusters of said cluster, whereupon the user may select one of these subclusters. The system locates the subdluster, via the appropriate pointer that was stored with the larger cluster, and allows the user to select one of its subdlusters from another menu. This process is repeated until the user comes to a leaf of the tree, which yields the details of an actual target object. Hierarchical trees allow rapid selection of one target object from a large set. In ten menu selections from menus often items (subdlusters) each, one can reach 10.sup.10 =10,000,000,000 (ten billion) items. In the preferred embodiment, the user views the menus on a computer screen or terminal screen and selects from them with a keyboard or mouse. However, the user may also

make selections over the telephone, with a voice synthesizer reading the menus and the user selecting subdlusters via the telephone's touch-tone keypad. In another variation, the user simultaneously maintains two connections to the server, a telephone voice connection and a fax connection; the server sends successive menus to the user by fax, while the user selects choices via the telephone's touch-tone keypad.

Detailed Description Text (324):
Users' navigational patterns may provide some useful feedback as to the quality of the labels. In particular, if users often select a particular cluster to explore, but then quickly backtrack and try a different cluster, this may signal that the first cluster's label is misleading. Insofar as other terms and attributes can pro vide "next-best" alternative labels for the first cluster, such "next-best" labels can be automatically substituted for the misleading label. In addition, any user can locally relabel a cluster for his or her own convenience. Although a cluster label provided by a user is in general visible only to that user, it is possible to make global use of these labels via a "user labels" textual attribute for target objects, which attribute is defined for a given target object to be the concatenation of all label s provided by any user for any cluster containing that target object. This attribute influences similarity judgments: for example, it may induce the system to regard target articles in a cluster often labeled "Sports News" by users as being mildly similar to articles in an otherwise dissimilar cluster often labeled "International News" by users, precisely because the "user labels" attribute in each cluster profile is strongly associated with the term "News." The "user label" attribute is also used in the automatic generation of labels, just as other textual attributes are, so that if the user-generated labels for a cluster often include "Sports," the term "Sports" may be included in the automatically generated label as well.

Detailed Description Text (333):
Although the topology of a hierarchical cluster tree is fixed by the techniques that build the tree, the hierarchical menu presented to the user for the user's navigation need not be exactly isomorphic to the cluster tree. The menu is typically a somewhat modified version of the cluster tree, reorganized manually or automatically so that the clusters most interesting to a user are easily accessible by the user. In order to automatically reorganize the menu in a user-specific way, the system first attempts automatically to identify existing clusters that are of interest to the user. The system may identify a cluster as interesting because the user often accesses target objects in that cluster -- or, in a more sophisticated variation, because the user is predicted to have high interest in the cluster's profile, using the methods disclosed herein for estimating interest from relevance feedback.

Detailed Description Text (334):
Several techniques can then be used to make interesting clusters more easily accessible. The system can at the user's request or at all times display a special list of the most interesting clusters, or the most interesting subclusters of the current cluster, so that the user can select one of these clusters based on its label and jump directly to it. In general, when the system constructs a list of interesting clusters in this way, the I.sup.th most prominent choice on the list, which choice is denoted Top(I), is found by considering all appropriate clusters C that are fuirther than a threshold distance t from all of Top(1), Top(2), . . . Top (I-1), and selecting the one in which the user's interest is estimated to be highest. Here the threshold distance t is optionally dependent on the computed cluster variance or cluster diameter of the profiles in the latter cluster. Several techniques that reorganize the hierarchical menu tree are also useful. First, menus can be reorganized so that the most interesting subcluster choices appear earliest on the menu, or are visually marked as interesting; for example, their labels are displayed in a special color or type face, or are displayed together with a number or graphical image indicating the likely level of interest. Second, interesting

clusters can be moved to menus higher in the tree, i.e., closer to the root of the tree, so that they are easier to access if the user starts browsing at the root of the tree. Third, uninteresting clusters can be moved to menus lower in the tree, to make room for interesting clusters that are being moved higher. Fourth, clusters with an especially low interest score (representing active dislike) can simply be suppressed from the menus; thus, a user with children may assign an extremely negative weight to the "vulgarity" attribute in the determination of q, so that vulgar clusters and documents will not be available at all. As the interesting clusters and the documents in them migrate toward the top of the tree, a customized tree develops that can be more efficiently navigated by the particular user. If menus are chosen so that each menu item is chosen with approximately equal probability, then the expected number of choices the user has to make is minimized. If, for example, a user frequently accessed target objects whose profiles resembled the cluster profile of cluster (a, b, d) in FIG. 8 then the menu in FIG. 9 could be modified to show the structure illustrated in FIG. 10.

Detailed Description Text (342):
1. The cluster profile for cluster C, or data sufficient to reconstruct this cluster profile. 2. The number of target objects contained in cluster C. 3. A human-readable label for cluster C, as described in section "Labeling Clusters" above. 4. If the cluster is divided into subclusters, a list of pointers to files representing the subclusters. Each pointer is an ordered pair containing naming, first, a file, and second, a multicast tree or a specific server where that file is stored. 5. If the cluster consists of a single target object, a pointer to the file corresponding to that target object.

Detailed Description Text (344):
The advantage of this distributed implementation is threefold. First, the system can be scaled to larger cluster sizes and numbers of target objects, since much more searching and data retrieval can be carried out concurrently. Second, the system is fault-tolerant in that partial matching can be achieved even if portions of the system are temporarily unavailable. It is important to note here the robustness due to redundancy inherent in our design--data is replicated at tree sites so that even if a server is down, the data can be located elsewhere.

Detailed Description Text (362):
Once Virtual Community Service identifies a cluster C of messages, users, search profiles, or target objects that determines a pre-community M, it attempts to arrange for the members of this pre-community to have the chance to participate in a common virtual community V. In many cases, an existing virtual community V may suit the needs of the pre-community M. Virtual Community Service first attempts to find such an existing community V. In the case where cluster C is a cluster of messages, V may be chosen to be any existing virtual community such that the cluster profile of cluster C is within a threshold distance of the mean profile of the set of messages recently posted to virtual community V; in the case where cluster C is a cluster of users, V may be chosen to be any existing virtual community such that the cluster profile of cluster C is within a threshold distance of the mean user profile of the active members of virtual community V; in the case where the cluster C is a cluster of search profiles, V may be chosen to be any existing virtual community such that the cluster profile of cluster C is within a threshold distance of the cluster profile of the largest cluster resulting from clustering all the search profiles of active members of virtual community V; and in the case where the cluster C is a cluster of one or more target objects chosen from a separate browsing or filtering system, V may be chosen to be any existing virtual community initiated in the same way from a cluster whose cluster profile in that other system is within a threshold distance of the cluster profile of cluster C. The threshold distance used in each case is optionally dependent on the cluster variance or cluster diameter of the profile sets whose means are being compared.

Detailed Description Text (363):

If no existing virtual community V meets these conditions and is also willing to accept all the users in pre-community M as new members, then Virtual Community Service attempts to create a new virtual community V. Regardless of whether virtual community V is an existing community or a newly created community, Virtual Community Service sends an e-mail message to each pseudonym P in pre-community M whose associated user U does not already belong to virtual community V (under pseudonym P) and has not previously turned down a request to join virtual community V. The e-mail message informs user U of the existence of virtual community V, and provides instructions which user U may follow in order to join virtual community V if desired; these instructions vary depending on whether virtual community V is an existing community or a new community. The message includes a credential, granted to pseudonym P, which credential must be presented by user U upon joining the virtual community V, as proof that user U was actually invited to join. If user U wishes to join virtual community V under a different pseudonym Q, user U may first transfer the credential from pseudonym P to pseudonym Q, as described above. The e-mail message further provides an indication of the common interests of the community, for example by including a list of titles of messages recently sent to the community, or a charter or introductory message provided by the community (if available), or a label generated by the methods described above that identifies the content of the cluster of messages, user profiles, search profiles, or target objects that was used to identify the pre-community M.